26 | 怎么规划和控制数据库的复制延迟大小?

2020-01-24 庄振运

讲述：庄振运
时长 21:55 大小 17.57M

你好，我是庄振运。

在正文开始前，首先要祝你新年快乐！今天我们要通过我在 LinkedIn（领英）公司做过的一个项目，来学习如何控制数据库复制延迟。

我在 LinkedIn 工作的时候，就遇到过因为数据复制延迟太大而导致的生产事故。当时出现了这样的情况：一个用户刚刚更新了自己的照片，可是他的朋友们却迟迟看不到更新。

你是不是觉得 LinkedIn 又不是微博，看个照片而已，早点晚点也没什么大不了的?

但换个内容，这性质就严重了。LinkedIn 网站上面会播放广告，在一定时间内，广告商往往会设置一定的广告投放预算限额，比如一天一万元。如果广告投放的收入统计数据被延迟，就会导致很严重的统计错误，实际支出严重地超过预算。一切有关钱的事，可都是大事。

所以，从用户数据的有效性、时间性和一致性来考虑，数据的传输复制延迟当然是越小越好。

什么是数据库的复制延迟?

那到底什么是数据库的复制延迟呢？如果让我用一句话来说，数据库的复制延迟其实就是当线上服务需要多个数据库时（比如为了分散流量），一条信息从源头数据库传递复制到下游数据库时经过的延迟。

还是用 LinkedIn 的例子来说明。LinkedIn 的全球用户已经超过 5.9 亿，用户流量异常庞大。这些用户流量，是由用户的各种活动而生成的事件数据。LinkedIn 的系统要求是，实时地捕获这些用户数据，并不断存储在后台数据库中。这些存放在数据库中的用户数据很重要，会被各种应用程序和其他服务读取，比如广告投放就需要消费这些数据。

在消费这些数据的时候，理论上来说，数据的消费者可以直接连接到后台数据库，去读取这些数据。但是，这些用户数据流具有大数据的几个特征，包括大规模和高可变性，并不适合直接连接后台数据库。因为，这会对后台数据库造成很大压力，而且不能保证读取的延迟。

为了解决这一挑战，当今主要的互联网公司，往往部署一整套的数据处理系统。这一整套系统通常由几个模块构成，包括数据事件的捕获、数据的存储、数据的复制传输、数据的读取。

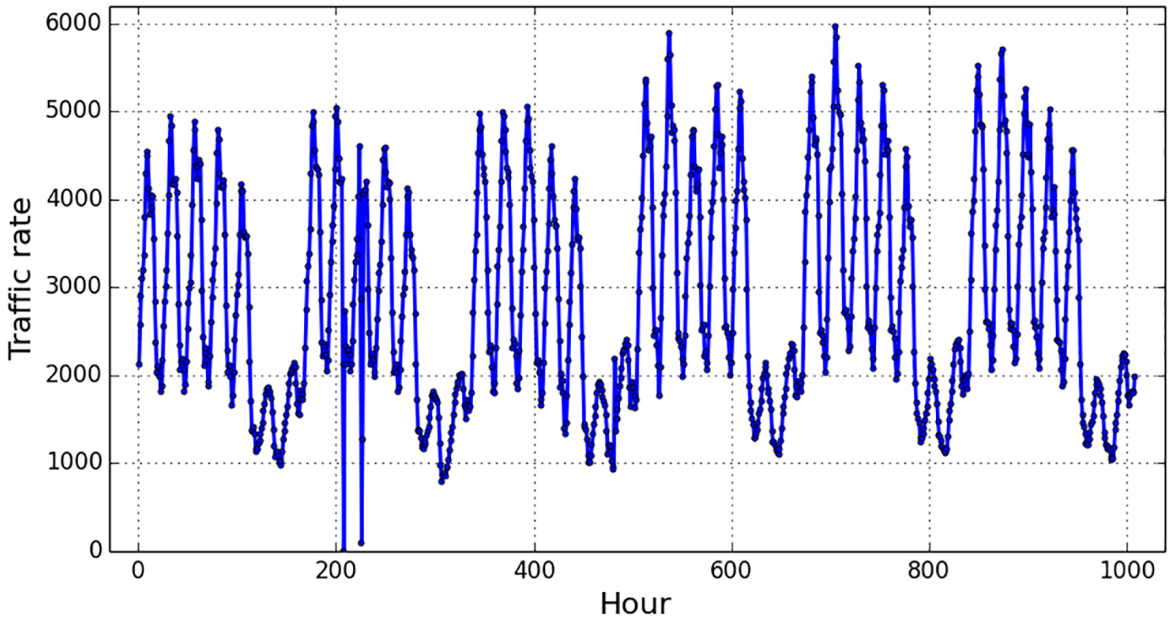
在这个系统中，**数据的复制传输**的作用很特别，是用来隔离数据库和数据使用者的。

采用数据复制传输模块的好处非常大，它可以减轻源头数据库的压力。因为复制传输可以进行级联，分散了用户流量，能够让系统的可扩展性更好。但这里有个前提，那就是，要保证数据的复制延迟不能太大，否则会造成很糟糕的业务影响，比如影响广告业务的收入。

我们今天就针对这个数据复制传输系统，来探讨如何通过合适的容量规划和分析，来控制数据的复制延迟。

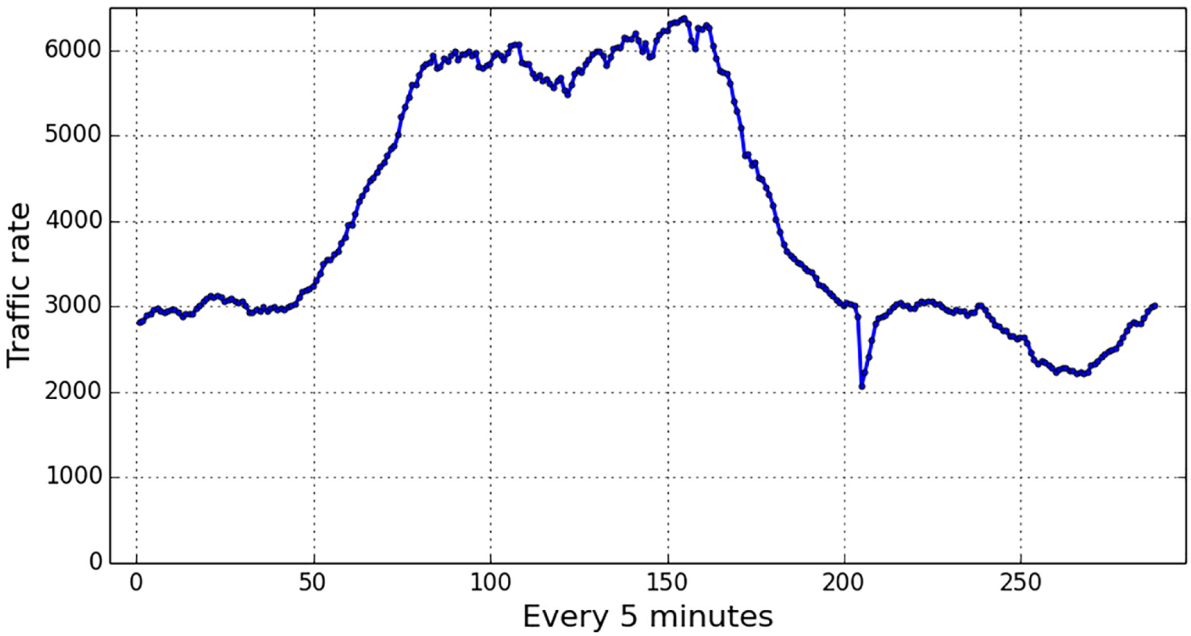
要想理解后面要讲到的容量规划，你得先去**了解生产环境中的用户流量特点**才行。

下面的图显示了一个互联网服务的在线用户流量。



时间范围是连续的六周，也就是 42 天的流量。你可以看到，用户流量呈现出了非常强的重复模式：基本上是以一周为一个周期，每周内五个工作日流量比较大，而周末的两天流量比较小。而且无论哪天，每天内部都有一个峰值。

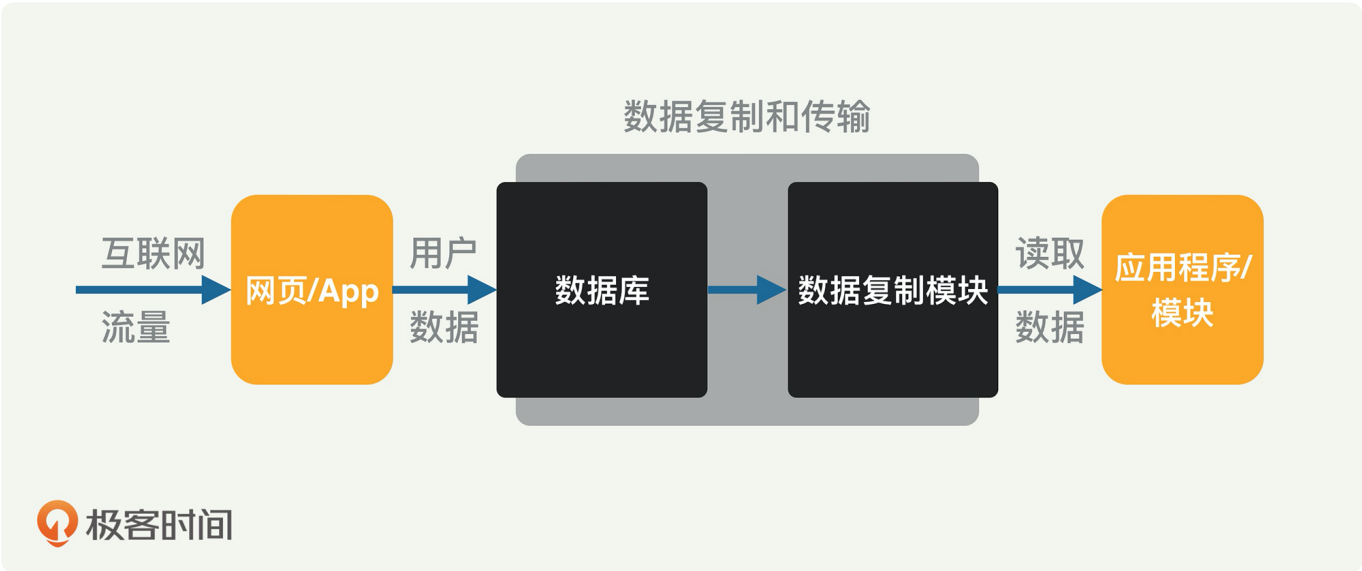
我们接着研究每一天内流量的周期性模式。对于每个工作日和周末，流量的形状也是非常有规律的曲线。下图显示了一个工作日中，也就是 24 小时内的流量变化。你可以清楚地看到，每天都会有一段时间的高峰期，约为 8 小时。



我们也对比了工作日和周末的流量变化，结果也是毫不奇怪的：工作日峰值一般远高于周末的峰值。二者的峰值之间，差不多有 4 倍的差距。

你只有了解了用户流量特征，才能在实际操作中选对模型。

了解了生产环境中的用户流量特征后，你还要了解 LinkedIn 采用的数据传输复制系统才行。如下图所示，LinkedIn 的系统有如下几个模块：事件生成、数据库存储、数据复制 / 传输和数据的消费 / 读取。

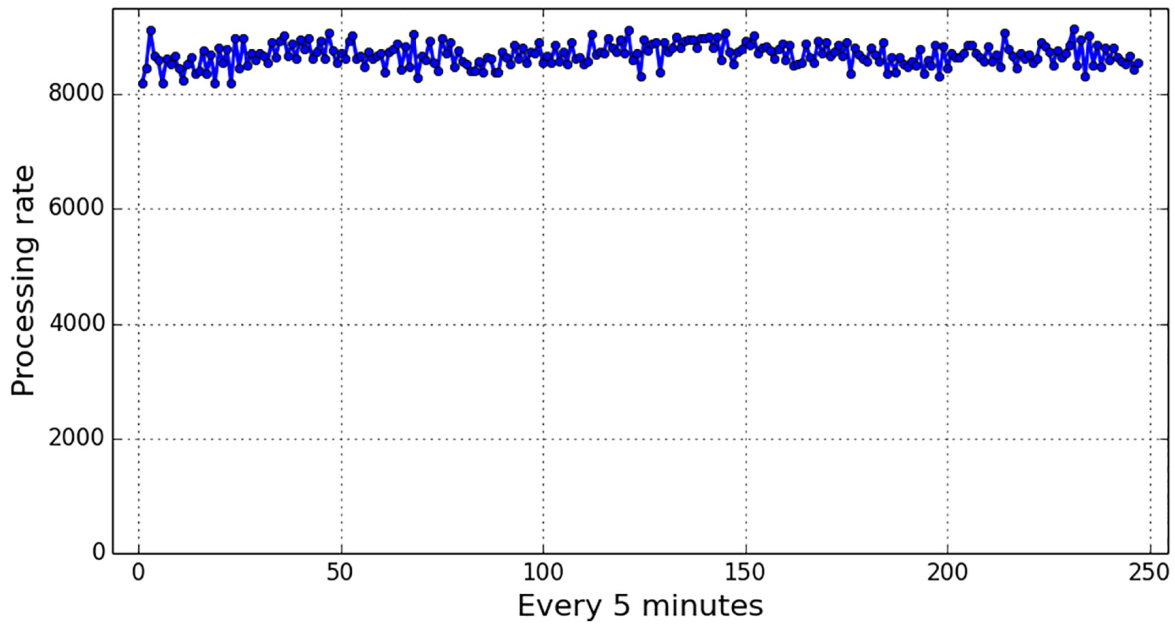


具体来讲，当用户与公司的网页互动时，相应的用户更新事件就被发送到了数据库。这些事件包括：用户点击了其他链接、阅读了其他用户的动态、向其他用户发信息等等。每个用户事件都由数据复制模块传输，并提供给下游消费者服务。

同一个用户数据，可能被很多应用程序和服务模块读取，所以这样上下游级联的系统设计，有比较好的扩展性，可以轻松应对规模的扩展。比如，如果数据复制模块不堪重负，而成为性能瓶颈，那么可以采用发散式级联的方式来扩展，从而分散读取的流量。

需要注意的是，**数据传输复制模块除了可以提供高扩展性，也可以提供数据的一致性**。像 LinkedIn 这样的公司，服务的用户遍布全球，在全球也就有很多数据中心。因为互联网流量分布在多个数据库或多个数据中心，所以就需要一个整合而一致的数据视图。这样一个目的，是可以通过传输复制实时数据库事件来得到的。

这个系统的另外一个特性是，**虽然用户流量可能随着时间变化很大，但是数据复制和传输的能力相对稳定**。我们通过测量和观察，发现复制传输模块的吞吐量能力非常稳定，如下图所示。



那么面对不断变化的客户流量（也就是事件流量），我们要如何规划整个系统，来让稳定的数据复制能力去适应用户流量的不稳定性呢？

怎么解决数据库复制延迟问题？

要达成目的，我们必须做好容量规划和分析。

要减少数据库复制延迟，我们在进行容量规划的时候需要考虑几个因素，包括**用户流量**、**复制模块的吞吐量**、**复制的延迟**以及**复制延迟的 SLA**（Service Level Agreements，服务水平协议）；从而确保所需的复制延迟，不要超过 SLA 的规定。

另外，通过充分考虑用户流入的流量和复制容量，我们还可以做出一些预测，比如未来一定时间内的预期数据复制延迟。而且，大多数互联网公司的流量往往有不断增长的趋势，我们需要持续地提高复制处理能力，来应对数据流量的增加。

数据库复制的容量规划和分析，其实就是在三个基本变量中辗转腾挪。哪三个变量呢？就是**用户流量大小**、**传输复制容量的能力**和**传输复制的延迟**。给定任何两个变量，都可以确定第三个变量。

具体来说，数据库复制的容量规划和分析可以帮助回答以下几个问题：

1. 未来的流量预测
2. 数据复制延迟的预测
3. 确定数据复制的容量
4. 确定用户流量的增长空间
5. 帮助确定延迟 SLA

未来的流量预测，也就是根据历史流量的数据，预期未来的流量（这个问题也能帮助回答后面的问题）。

数据复制延迟的预测，也就是给定传入流量和复制处理能力，预期的复制延迟是多少？这些数据，可以帮助我们确定复制延迟的 SLA。

确定数据复制的容量，就是在假设给定传入流量和最大允许的复制延迟（SLA）的情况下，确定我们需要部署多少数据复制容量？这将有助于定义复制容量需求。

确定用户流量的增长空间，就是在给定复制的容量和延迟的 SLA 的情况下，确定最大可以支持多少用户流量？这有助于计划将来的容量要求。

帮助确定延迟 SLA，就是在给定输入流量，现在或将来的复制处理能力的情况下，如何确定适当的 SLA？显然，作为一个公司或者部门，我们不想过度承诺或低估 SLA。

解决方案如何落地？

那么这具体是如何操作的呢？现在我来为你介绍一下我们采用的规划和分析模型（这是基于统计模型的）。

你要知道，数据的传输复制模块，其实是一个排队系统，是一个**有无限缓冲的先进先出队列**。

从输入角度讲，所有的用户流量数据都进入这个先进先出队列，然后传输复制模块会一个一个地处理。如果在任何时候，用户流量大小超过传输能力，那么队列就会加长。反之，队列就会变短。不过与典型的排队论问题不同，在这里，我们会更加**关注所有事件的“最长”等待时间，并根据这个值来决定 SLA**。

你还记得我们在最开始对用户流量特征的观察吗？基于前面对流量的观察和测量，你认为该用什么模型来预测未来数据呢？

没错，最好把用户流量看作是一个时间序列模型。并通过这个模型来预测未来的数据。

对于这类时间序列数据，通常选择 **ARIMA**（Autoregressive Integrated Moving Average，自回归积分移动平均线）模型，来进行建模和预测。什么是 ARIMA 呢？ARIMA 是一个常见的统计模型，是用来对时间序列进行预测的模型。它的全称是自回归移动平均模型（ARIMA, Autoregressive Integrated Moving Average Model)。

ARIMA 的工作机制有较大的计算开销，所以 ARIMA 不太适合大规模的建模，比如超过几百个数据点就不太合适了。一般来讲，对于一年内的预测，ARIMA 只能预测到每天的尺度，因为一年只有 365 天。对这个模型来讲，比天更细的粒度，不太适合。但是我们这里的容量规划，又偏偏需要获取更加精细粒度的数据，比如每小时的预测数据，而不仅仅是每天的数据。

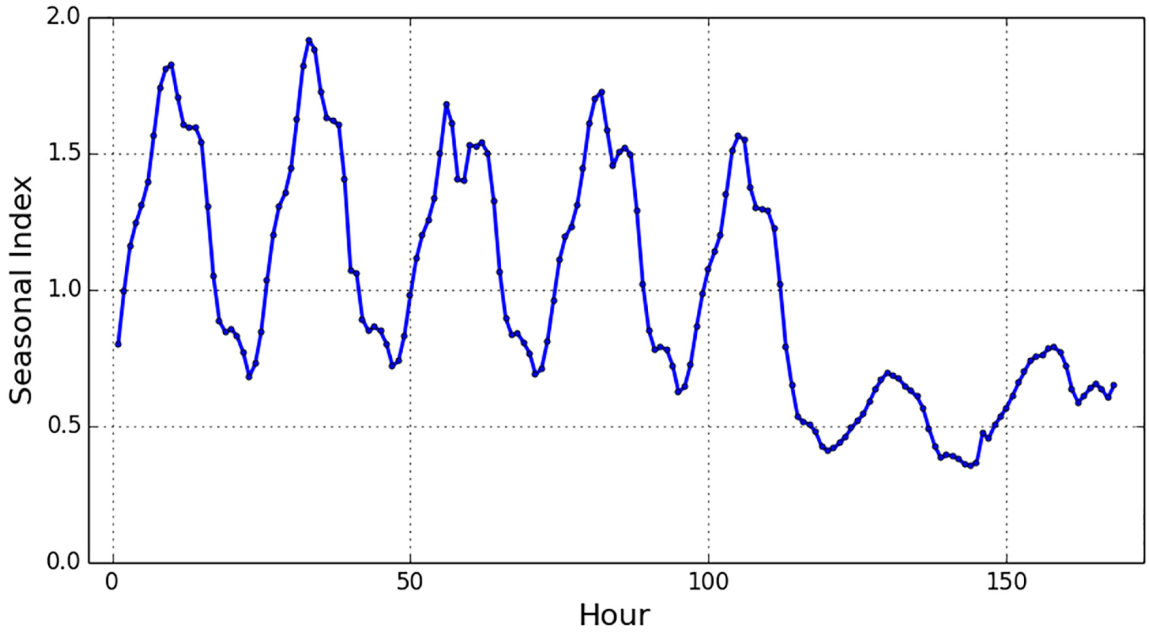
考虑到这几个特点，对于较长期的预测，比如半年期间的话，该怎么办呢？我们提出了**两步预测模型**来获取未来的每小时流量。

简而言之，这个两步预测模型采取两步走的办法。

第一步是工作在**星期**这个粒度上，获取每周的聚合流量；第二步再将聚合数字“分布”（或“转换”）到一天内的每个小时。这样做的好处是，一方面减少了数据的点数，比如一年也就 52 周；另一方面仍然可以得到每小时的数据。

你可能想问，这个第二步的转换，也就是流量从一周到小时的“转换”，是怎么做的呢？事实上，我们采用了一个**季节性指数**，该指数大致代表一周内每小时的流量部分。你也可以直观地把这个指数看作是一个分配函数。

下图就是一周到小时的季节指数展示。一周内有 168 小时，也就是有 168 个数据点。这些数据点的值大小，其实就是分布概率。所有的数据点的值加在一起正好等于 1。



具体来说，该模型包括两个步骤：

1. 使用 ARIMA 模型，预测未来几周的每周总体流量；
2. 使用季节指数，预测一周内每个小时的流量。

采用这个模型就可以回答我们前面提出的问题了，比如可以轻松地预测未来每小时的流量。

那么具体的算法怎么实现呢？比如回答与容量规划相关的其他几种类型的问题，我们采用了**数值计算**和**二进制搜索**的类似机制。

这里的二进制搜索原理，不难理解，其实就是不断地尝试，直到找到一个最合适的值为止。比如我们需要一个确定总的容量，一方面希望容量越小越好（因为省钱）；另一方面又不能违反传输复制延迟的 SLA。

怎么用二进制搜索呢?

就是随便假定一个容量值，然后带入模型去推导出传输复制的延迟，然后判断这个延迟是否违反了 SLA。如果违反了，说明我们一开始的容量值太小，应该增加一点。怎么寻找呢，就是每次取中间的值，重新推导。这个过程如同二进制搜索。

为了便于理解，我们假设时间粒度为每小时。一旦获得流量，则将使用数值计算方法，来判断任何时间点的复制延迟。基于数值计算的结果，也就可以使用二进制搜索，来获得特定流量所需的数据复制容量。

同样的道理，还可以用此模型，来确定任何固定场景可以支持的最大用户流量，以及相应的将来日期。也就是根据预测的增长，未来什么时候用户的流量就会超越这个最大用户流量。

最后，检查每种场景下的预期复制延迟，我们还可以确定适当的复制延迟 SLA。

任何针对流量的预测模型，都不可避免地会出现误差。产生预测误差的主要原因是网络流量的高变化性。

当实际流量小于预测流量时，实际的复制延迟也将低于预期延迟。注意，这种低估误差不会违反基于预测值定义的延迟性能 SLA。唯一付出的成本，是部署了一些额外的资源。但是，当预测的流量小于实际流量时，就可能会违反延迟 SLA。

为了解决此问题，在确定各种指标（例如 SLA）时，有必要**预留一定的空间来避免预测误差**。预留的量取决于一系列因素，包括：

1. 预测模型的历史表现，也就是预测的精准度；
2. 违反延迟 SLA 的后果，包括业务付出的成本；

3. 过度部署容量资源的成本。

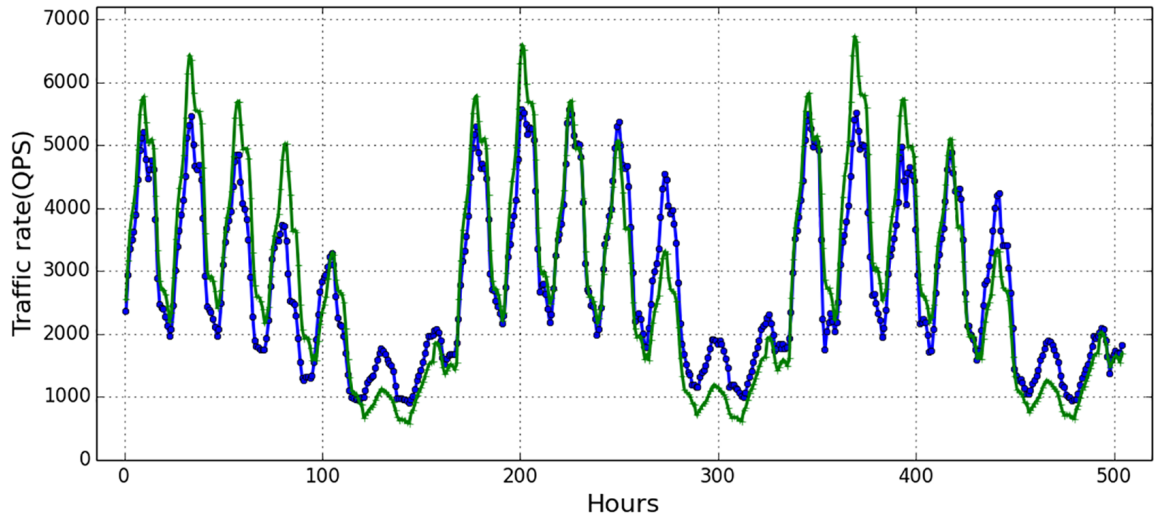
好了，到这里这个规划和分析模型就讲完了，内容比较多，我们来复习一下。对于数据库复制系统来说，可以把它当作一个排队系统来对待。不断到来的输入流量，是一个有季节性的不断变化的时间序列。通过对这个时间序列来适当建模，我们可以比较准确地预测未来的输入流量。有了预测的输入流量，就可以根据服务处理的速度和复制延迟的关系，来计算出其他结果：比如预期的复制延迟，所需要的复制容量等等。

实际生产实践的验证

我们在 LinkedIn 的生产实践中，采用了这个规划，下面我们来看看一些结果。

我们还是使用前面介绍的 42 天数据，来做未来的流量预测。使用 ARIMA 模型的实践建议是，预测数据的长度最好不要超过历史数据长度的一半。所以，既然我们有 42 天的历史数据，通常可以在未来的 21 天之内，使用 ARIMA 模型进行预测。

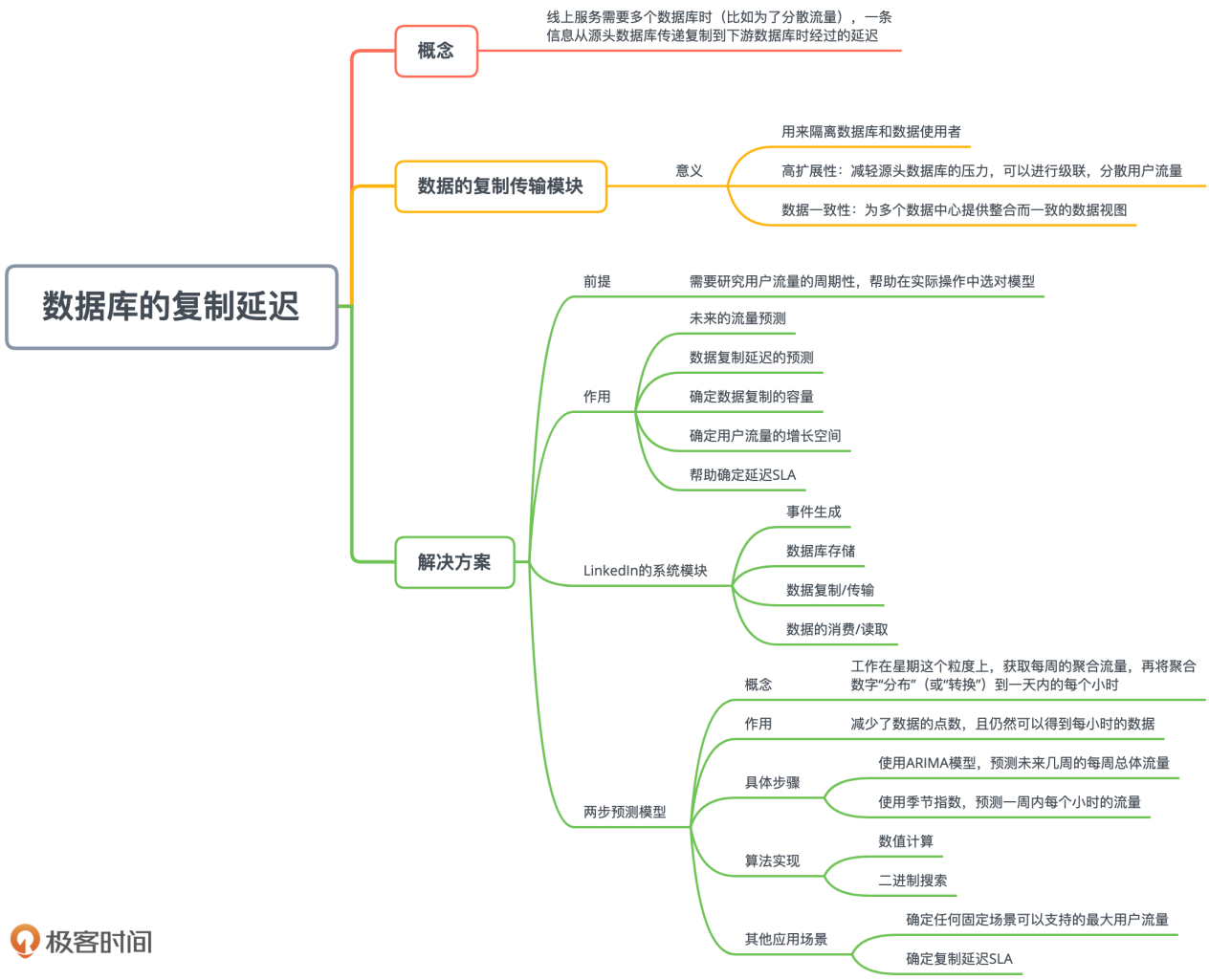
下图展示了这 21 天的两个数据，一个是生产环境中的实际数据，另外一个是根据我们模型的预测值。蓝颜色的线条是实际观察到的用户流量，绿颜色的线条是我们的预测值。



从图中可以观察到，虽然有些地方还是有误差，但是从解决实际问题的角度，我们认为已经足够了。我们也把 ARIMA 方式和其他预测方式做了比较，比较预测的时间序列结果，我们发现 ARIMA 模型给出了更好的精度；但差异不大，仅仅有 6%左右的差异。

总结

唐代诗人杜甫有两句诗说：“迟日江山丽，春风花草香。”说的是春天越来越长，沐浴在春光下的江山格外秀丽，春风也送来花草的芳香。数据传输复制的延迟，如果不仔细控制，也会变得越来越长，最终会导致严重问题。到时候就不是春风送暖和花草飘香，而是秋风萧瑟，老板发威了。



我们今天探讨了如何通过合理的规划和分析，来控制这个数据库复制延迟。具体就是使用 ARIMA 统计模型，并且用两步走的策略，来回答一系列的各种相关问题。

我想特意说明的是，这一整套时间序列的预测原理和方法，其实也可以用在很多其他的很多种预测场景中。比如一个在线广告显示系统（输入的流量是线上用户活动量，要进行的处理是决定合适的广告来显示）。

通过今天的分享，我希望你能掌握的要点是对于一个线上的数据处理系统，我们可以进行基于排队理论和时间序列的建模，通过这个建模来回答各式各样的相关问题。这个方案其实还牵涉到一些稍微复杂的统计理论，如果你还有精力，我建议你去读一下我的一篇论文，发表

在ACM ICPE上面，里面阐述了所有的细节，你在阅读的过程中有什么问题或者思考，也可以留言和我讨论。

思考题

如果数据的传输复制延迟过大，会造成很多种不同场合的业务影响，都会有什么样的业务影响呢？

提示：假如你们公司的网站上面帮助别人投放广告，如果广告的数据有延迟，会造成广告商和你们公司之间的什么样的纠纷呢？

欢迎你在留言区分享自己的思考，与我和其他同学一起讨论，也欢迎你把文章分享给自己的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　25 | 如何在生产环境中进行真实的容量测试？

下一篇　27 | 多任务环境中的Java性能问题，怎样才能不让程序互相干扰？

精选留言 (4)

写留言

hello
2020-01-24

老师，祝您新年快乐

展开

作者回复: 谢谢！新年快乐！

2

kyleqian
2020-01-31

请问LinkedIn是用的mysql数据库吗？对于单个数据中心，对于大量用户事件写入和读取怎么解决的？是通过数据库复制做主从集群吗？我觉得主库估计也得做分片吧？目前来看解决这个复制延迟主要就是通过部署更多节点的方式吗？

展开

黄海峰
2020-01-28

听不懂，应该具体一点，实践篇应该具体到用什么软件什么命令执行，现在实践篇也是理论。。。

Scott
2020-01-27

这个好像对我们很有用，我们预估是直接用上一周同一天的流量来预测，问题是有时候遇到特殊情况，比如春节国庆，就行不通了。

展开

作者回复: 嗯，如果是面向终端客户的，特殊日期/事件必须单独考虑。